TITLE: CR80 AMOS, FILE NAME UTILITIES
PRODUCT SPECIFICATION

DOCUMENT NO: CSS/317/PSP/0014

PREPARED BY: Flemming Pedersen

APPROVED BY: Jørgen Høg

AUTHORIZED BY: Jørgen Høg

DISTRIBUTION:

ISSUE:	1								
DATE:	791115								

CR80 AMOS, FILE NAME UTILITIES, PRODUCT SPECIFICATION	sign/date FP/791115	page
	repl	project

PAGE ISSUE RECORD AND CHANGE LOG

PAGE	ISSUE							
	1	2	3	4	5	6	7	8
01								
02								
03								
04								
05								
06								
07								
08								
09								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								
37								
38								
39								
40								
41								
42								
43								
44								
45								
46								
47								
48								
49								
50								
51								
52								
53								
54								
55								
56								
57								
58								
59								
60								
61								
62								
63								
64								
65								
66								
67								
68								
69								
70								
71								
72								
73								
74								
75								
76								
77								
78								
79								
80								
81								
82								
83								
84								
85								
86								
87								
88								
89								
90								
91								
92								
93								
94								
95								
96								
97								
98								
99								
100								

ISSUE	DATE	PREPARED BY	APPROVED BY	AUTHORIZED BY
1	791115	FP.		

TABLE OF CONTENTS

1. SCOPE

The purpose of this document is to describe the functions performed by the AMOS file name utility package.

The file name utility package is identified as configuration item CSS/317.

2. APPLICABLE DOCUMENTS

The following documents contain reference information for the present document.

2.1 CR FILE SYSTEM PSP
CSS/910/EWP/0001

2.2 CR80 AMOS, I/O SYSTEM
PRODUCT SPECIFICATION
CSS/006/PSP/0006

2.3 CR80 AMOS, DIRECTORY UTILITY PACKAGE
USER's MANUAL
CSS/932/USM/0036

3. FUNCTIONAL DESCRIPTION

This section describes the functions performed by the two routines. INFILEID and FINDFILE which constitute the file name utility package.

INFILEID reads and interpretes a fileid according to the standard AMOS syntax (ref. 2.3) from a specified stream. The fileid is delivered in parameters which may be used for a succeeding call to FINDFILE.

FINDFILE looks a file up according to a fileid as specified by the parameters delivered by INFILEID.

3.1 Activation

The file name utility package should be part of an AMOS boot module.

The package is implemented as monitor procedures, and is therefore activated at system initialization time to initialize the entries in the monitor jump table.

The routines of the file name utility package is invoked from an application program by means of MON instructions.

The procedures INFILEID and FINDFILE are further available to application programs through the standard PASCAL prefix.

3.2 Parameter Declarations

Below is given PASCAL-like declarations of the parameter utilized by INFILEID and FINDFILE as called from assembly code programs. Also, refer to ref. 2.2.

```
FILE_SYSTEM_NAME: RECORD
                  PROCESS_NAME: RECORD
                                 NAME:   ARRAY [0..5] OF BYTE;
                                 IDENT:  INTEGER
                                END
                  GROUP_NAME:   ARRAY [0..3] OF BYTE
                  END;

VOLUME_NAME: ARRAY [0..15] OF BYTE;

NAMELIST:    ARRAY [1..10] OF ARRAY [0..15] OF BYTE;

WORKAREA:    ARRAY [1..16] OF INTEGER;

NAME_NO:     INTEGER;

FROM_ADAM:   INTEGER; "USED AS BOOLEAN"
```

3.3 Function of INFILEID

INFILEID reads and interpretes a file id from a specified stream. The stream must be connected to a file for input, and it must be positioned to a letter, a '@', or a blank. INFILEID calls the AMOS I/O SYSTEM routine INELEMENT to read the next syntactic element from the stream. This implies that the standard line continuation sign ('%') may be used.

If the first character read is a blank, a string of blanks is skipped. If the first non-blank character read is a '@', FROM_ADAM is set to TRUE (=#FFFF), otherwise to FALSE (=0). IF a FILE_SYSTEM_NAME, a VOLUME_NAME, or both are present, they are delivered in the corresponding parameter. Otherwise these parameters are left unaltered. This means, that appropriate defaults for FILE_SYSTEM_NAME and VOLUME_NAME should be assigned to the corresponding parameters prior to the call of INFILEID; at least if the parameters are intended for direct use by FINDFILE.

FILE_NAMEs are delivered (to a maximum of 10) in the array FILENAMELIST, together with the number of FILE_NAMEs present, which is delivered in NAME_NO.

If an I/O error occurs during read, INFILEID will return to LINK with the I/O completion code in R7.

If a syntax error is recognized, INFILEID will return to LINK+1 with "syntax error" (=#218) in R7. The stream position will not be changed after the syntax error has been detected.

If no errors are found, INFILEID will return to LINK+2 with "io-ok" (=0) in R7. In this case the stream will be positioned at the first character which is not part of the fileid.

3.4 Function fo FINDFILE

FINDFILE searches through a directory chain for a named file. The fileid is specified by parameters as delivered by INFILEID.

If FROM_ADAM is true a GET_ROOT command is performed with FILESYSTEMNAME and VOLUMENAME. Otherwise, the first name in the NAMELIST is used for a LOOKUP command on the specified DIRectory. The remaining (up to NAME_NO) names of the NAMELIST are used for DESCENT commands from the (supposed) directory obtained from GETROOT or LOOKUP. If FROM_ADAM is true and NAME_NO equals zero, the root directory of the specified volume of the specified filesystem is returned. If FROM_ADAM is false and NAME_NO equals zero, the specified DIRectory is returned unaltered.

If a LOOKUP or DESCENT is applied to a file which is not a directory, FINDFILE will return a COMPLETION_CODE of ILLEGALDIRECTORY (or perhaps PROTECTION_FAILURE).
If FROM_ADAM is true, the completion codes "unknown file system" and "nonexisting volume" may be returned. In case of an error, FINDFILE will return to LINK+0, with the completion code in R7. If no errors are found, FINDFILE will return to LINK+1.

3.5 Call Specifications

3.5.1 Assembly Code

The application program must supply a work area of at least 16 words referenced by R6, to both INFILEID and FINDFILE. This work area must lye inside the process area, and the reference must be a process base relative word address.

3.5.1.1 INFILEID

Invocation:

```
MON INFILEID
```

Register Use:

	CALL:	RETURN:
R0:	REF(FILESYSTEM NAME)	REF(FILESYSTEM NAME)
R1:	REF(VOLUME)	REF(VOLUME)
R2:		NAME_NO
R3:		FROM_ADAM
R4:	STREAM	STREAM
R5:	REF(NAMELIST)	REF(NAMELIST)
R6:	REF(WORKAREA(16WORDS))	DESTROYED
R7:	LINK	COMPLETION CODE

Return Points:

LINK+0:	IO ERROR
LINK+1:	SYNTAX ERROR
LINK+2:	DONE

3.5.1.2 FINDFILE

Invocation:

```
MON FINDFILE
```

Register Use:

	CALL:	RETURN:
R0:	REF(FILESYSTEM NAME)	REF(FILESYSTEM NAME)
R1:	REF(VOLUME)	REF(VOLUME)
R2:	NAME_NO	NAME_NO
R3:	FROM_ADAM	FROM_ADAM
R4:	DIR	FILE
R5:	REF(NAMELIST)	REF(NAMELIST)
R6:	REF(WORKAREA(16WORDS))	DESTROYED
R7:	LINK	COMPLETION CODE

Return Points:

LINK+0:	ERROR
LINK+1:	DONE

3.5.2 PASCAL Prefix Declarations

3.5.2.1 Types

```
   TYPE PROCESS_NAME = RECORD
                           NAME: ARRAY [0..2] OF INTEGER;
                           NAME_IDENT: INTEGER
                       END;

   TYPE FILE_SYSTEM_NAME = RECORD
                               PNAME: PROCESS_NAME;
                               GNAME: ARRAY [0..1] OF INTEGER
                           END;
   TYPE PACKED_NAME = ARRAY [0..7] OF INTEGER;
   TYPE VOLUME_NAME = PACKED_NAME;

   TYPE FILE_NAME = PACKED_NAME;

CONST NAMELISTMAXINDEX = 10;
TYPE NAMELISTTYPE = ARRAY [1..NAMELISTMAXINDEX] OF PACKED_NAME
```

3.5.2.2 INFILEID

```
   PROCEDURE INFILEID(S: STREAM;
                       VAR FROM_ADAM: BOOLEAN;
                       VAR FSN: FILE_SYSTEM_NAME;
                       VAR VOLUME: VOLUME_NAME;
                       VAR NAMELIST: NAMELISTTYPE;
                       VAR NAME_NO: INTEGER;
                       VAR CC: COMPLETION_CODE);
```

3.5.2.3 FINDFILE

```
   PROCEDURE FIND_FILE(FROM_ADAM: BOOLEAN;
                       FSN: FILE_SYSTEM_NAME;
                       VOLUME: VOLUME_NAME;
                       NAMELIST: NAMELISTTYPE;
                       NAME_NO: INTEGER;
                       DIRECTORY: FILE;
                       VAR F: FILE;
                       VAR CC: COMPLETION_CODE);
```